Royal Military College of Canada

EEE 455 Engineering Design Project

DID-07 Detailed Design Report - SAR Imaging Platform

Authors
OCdt Carter Green
NCdt Royce Burningham

Supervisor
Dr. Joey Bray
PMO
Maj. John Lloyd

April 7, 2020

Contents

List of Figures

Abbreviations

FMCW Frequency Modulated Continuous Wave.

FOV Field of View.

GPIO General Purpose Input and Output.

HPBW Half-Power Beam Width.

PD pulsed Doppler.

SAR Synthetic Aperture Radar.

SFTP Secure File Transfer Protocol.

SOR Statement of Requirements.

1 Introduction

1.1 Document Purpose

The purpose of this document is to provide a detailed insight into the design, architecture, and results of the Synthetic Aperture Radar (SAR) imaging project. This document will present:

(a) the aim and scope of the project

(b) a description of how the project functions

(c) the modules or flow that the code goes through

(d) the equipment used

(e) how tests were conducted

(f) the results of said tests

1.2 Background

SAR imaging is the creation of two and three-dimensional images using a radar and signal processing techniques. The idea of a SAR image has existed since the 1950s [8]. SAR images ability to see through weather conditions that impact visibility (snow, rain, fog, etc.) make SAR images a useful tool to have aboard most airborne surveillance systems [8]. However, in the world of SAR imaging, pulsed Doppler (PD) radars have dominated due to their advantages at long range when compared to Frequency Modulated Continuous Wave (FMCW) radars[8]. As a result, the use of an FMCW radar in the same situation has not been as established as that of PD. The lack of extensive research on FMCW SAR images makes FCMW's usefulness compared to PD an area of research interest. Moreover, SAR images use has been heavily focused on airborne systems due to the requirement to travel large distances to synthesize a coherent image, and the use of ground systems has not been fully explored. SAR imaging is of substantial interest to Armed Forces as low-cost, high utility surveillance platforms aid them in the gathering of intelligence. The lack of testing of FMCW SAR imaging, as well as SAR's value to the military, make a ground-based FMCW surveillance tool a perfect research area in which to test for viability.

1.3 Project Aim

The goal of the project is to utilize an FMCW radar and transform it into an imaging system by implementing SAR image processing techniques. FMCW's higher frequency and lack of extensive use in this field provided a way to implement SAR imaging in a novel way while still providing a low-cost imaging system with all the benefits that come along with SAR imaging. Along with demonstrating SAR imaging in a new context, the SAR imaging system is to be mounted on a mobile platform in order to demonstrate how a SAR imaging system could be implemented on a ground-based system.

1.4 Scope

The scope of the project was to develop an image by implementing a SAR imaging algorithm on a mobile platform. The platform is to wait until a start command is sent by the user on a laptop from a remote location. The platform was to travel over a set distance in discrete steps while simultaneously taking measurements and storing the data taken from the radar. After travelling the set distance, the platform was to form an image based on the taken measurements and send the resulting image back to the laptop that initiated the sequence.

2 Design Overview

This section is an overview of the techniques which are used in the project, a description of how the sequence works from beginning to end, as well as the explain the equipment used to create the project.

2.1 Synthetic Aperture

Given an antenna array, such as the array in Figure 1, with equal separation d, constant amplitude, and a linear phase progression Ψ, one can model the radiation pattern of an array. Equation 1 is the argument of the array factor, f_{Anorm} (eq. 2), which represents the radiation pattern of the array. Each antenna has an element factor which has a radiation pattern that is determined by the physical properties of the antenna. The element factor of a DemoRad antenna can be seen in Figure 2. The product of the array factor and the element factor gives the overall pattern of the antenna array. Multiplication of the radiation patterns is equivalent to a spatial convolution of the antenna and the arrays. The platform will only be moving in a straight line so the antenna arrangement is the convolution of the antenna and the locations of the antennas.

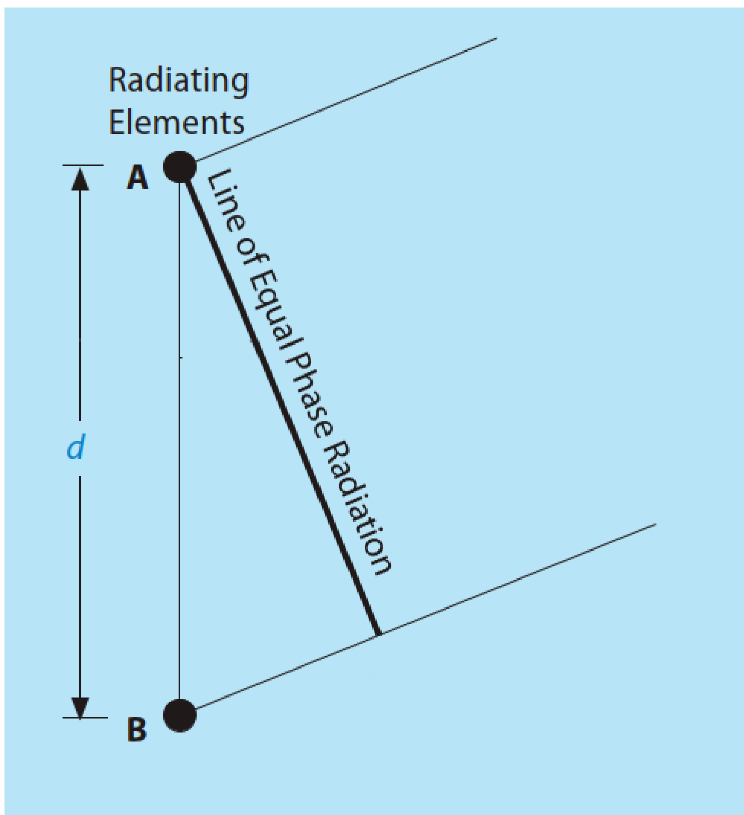


Figure 1: 2 Element Antenna Array [8]

$$\psi = \beta d \cos\theta + \Psi \tag{1}$$

$$f_{Anorm}(\psi) = \left| \frac{\sin\frac{N\psi}{2}}{N \sin\frac{\psi}{2}} \right| \tag{2}$$

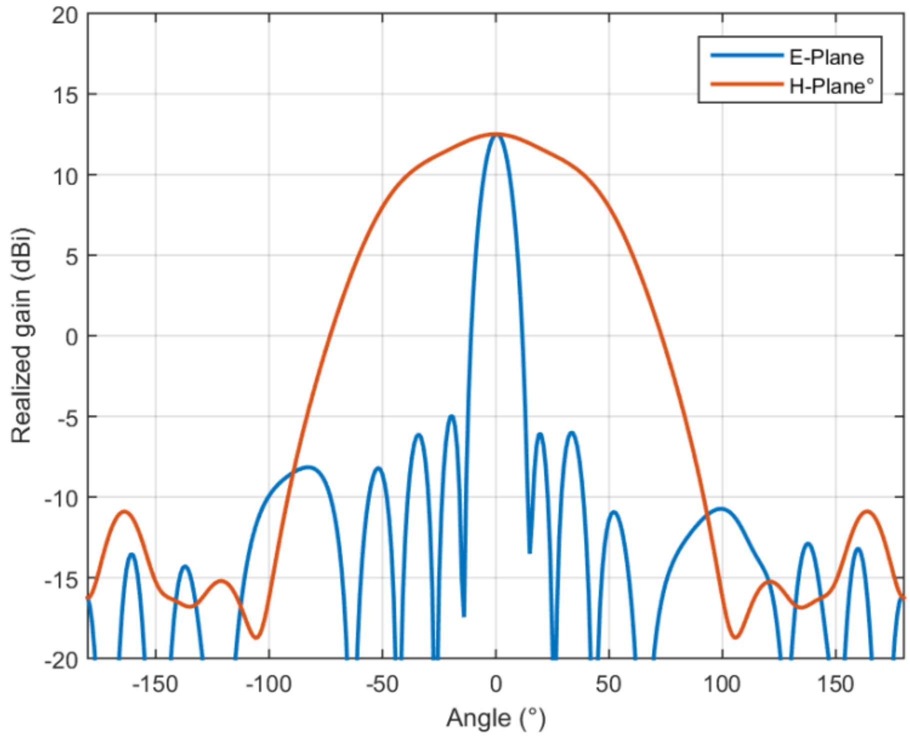


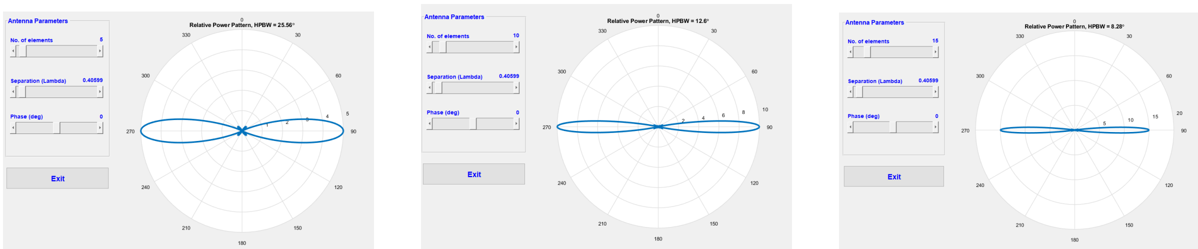
Figure 2: Radiation Pattern of a Single DemoRad Transmit Antenna [5]

Figure 3 shows the power pattern ($|f_{Anorm}(\psi)|^2$) of a linear array with no linear phase progression ($\Psi = 0$) and separation of 0.40599λ. Figures 3a-c show the radiation pattern of an antenna array as the number of antenna elements, α, is increased from 5 to 15 elements. Figure 3 shows that as the number of antennas increases in the array the Half-Power Beam Width (HPBW) is reduced. In SAR imaging, measurements are taken at different points and stored. When these individual measurements are added together the new measurement is what would be seen by a real array where HPBW of the resulting signal will be thinner just like a physical array.

Equation 3 provides the relationship between the cross-range resolution and the length of the synthetic antenna array. D_{SAR} is equal to Nd the number of elements multiplied by the separation. The resolution will improve proportionately as the number of elements is increased. However, increasing the number of elements is a trade off between resolution and computational complexity.

$$\Delta CR = \frac{\lambda R}{2D_{SAR}} \tag{3}$$

Also important to the radiation pattern given by equations 1 and 2 is the spacing between elements, d. When the spacing exceeds $\lambda/2$, side lobes that are equal in magnitude to the main lobe appear in the radiation pattern. In Figure 4, the main lobe is at 0° and grating lobes appear at equal magnitude at 90° and 270°. This is undesirable in SAR imaging as these side lobes will contain data from two different directions which will distort the desired data in the main lobe. An analogue to this in optical imaging would be if you took a picture and it had objects in a different direction overlaid with the subject of the picture in front of the camera.



(a) 5 Element Array with HPBW=25.56°

(b) 10 Element Array with HPBW=12.6°

(c) 15 Element Array with HPBW=8.28°

Figure 3: Power Patterns for an Antenna Array with a separation of $< 0.40599\lambda$ where 90°is broadside to the antenna

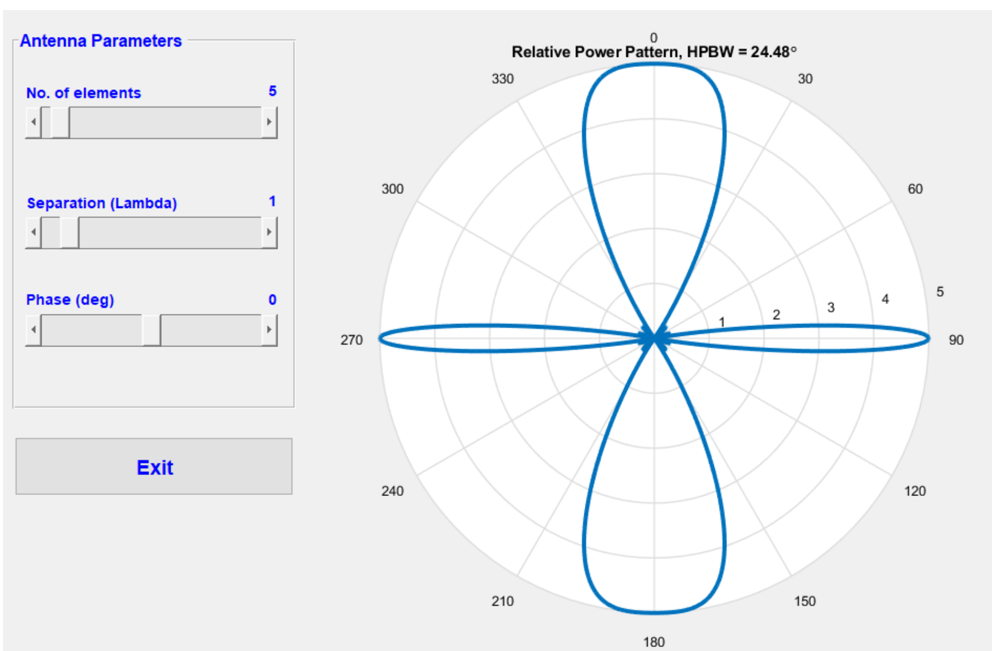


Figure 4: Appearance of Grating Lobes when $d > \lambda/2$

2.2 Movement Restrictions

Due to the aforementioned grating lobes, a worst-case scenario restriction was placed on the movement where each movement to a new measurement position could not exceed the distance given by equation 4. This equation gives a restriction to the separation of elements, d, due to the wavelength of the transmitted frequency λ and the HPBW of the radar θ_{3dB} in radian (76.5° or 1.335 radians[5]).

$$d < \frac{\lambda}{2(\theta_{3dB})} \tag{4}$$

The wavelength can be determined using the following universal wave equation where f is the centre frequency of the radar (24.1 GHz) and c is the speed of light ($3e8$ m/s). Using the wave equation, the wavelength of the radar was found to be 1.24cm

$$c = \lambda f \tag{5}$$

Using this information in equation 4 the platform should move less than 4.66mm between measurements to meet the separation restrictions.

2.3 Strip-map SAR Algorithm

This section will explain how the SAR imaging is implemented.

The DemoRad is configured to take a measurement of 256 samples which are stored in memory as the DemoRad creates the synthetic array.

Let:

d_0 — be the distance that one step of a stepper motor moves
d_u — be a user input of how far the platform will move to image
N — be the number of steps required by the stepper motor to move a d_u
α — be the number of synthetic elements to sum in the synthetic array
x — be the input signal to the SAR algorithm measured by the DemoRad with dimensions 256 x N
w — be a Hanning window with the same dimension as a single measurement (256 x 1)
Y — be the complex-valued output signal to be represented as an image

The program starts with the user inputting the distance δ_n that the platform will image. From this the amount of steps that the motor needs to move is calculated by equation 6

$$N = \left\lceil \frac{d_u}{d_0} \right\rceil \tag{6}$$

Then the platform will move the distance given by the user, d_u and take measurements every step of the motor at an interval δ_n. Each measurement is multiplied by a Hanning window to smooth the collected data. Once all measurements have been collected a discrete Fourier transform will be performed on each radar measurement to turn the measurements into range data. The discrete Fourier transform gives a symmetric return of frequency data because x is a real signal. Due to this symmetry, we remove all negative frequency data from the output of the transform giving the new complex-valued signal, X, half the length of the measured signal. Because the radar is an FMCW radar, the discrete frequencies of the transformed signal give the range of an object by equation 7 where k_f is the rate that the radar transmitter increases its frequency measured in Hz/s.

$$R = \frac{cf}{2k_f} \tag{7}$$

To create the output image, α radar measurements are taken and the values at each range, i, are summed to output a column, j. This column represents one column of pixels of the output image. This is repeated by adding a new measurement to the end of the synthetic array and removing the measurement from the array. The previous operation will be completed $N - \alpha$ times to form the image. The output can be represented by equation 8. To view the image, the magnitude of the complex signal Y is taken and plotted using the pcolormesh function from the Python library matplotlib.

$$Y[i,j] = \sum_{n=0}^{\alpha} X\big[i, j-n\big] \tag{8}$$

Where, $X\big[i,j\big]$ is the output of the discrete Fourier transform of x

The above SAR algorithm describes unfocused SAR. The output of this algorithm is unfocused because it does not account for the difference in distance from a synthetic element to a target. A SAR image can be focused by adding a phase shift to account for the different distances from the synthetic elements to the target. The difference in distance to a target that causes a phase shift in the measured symbol is seen below in Figure 5.

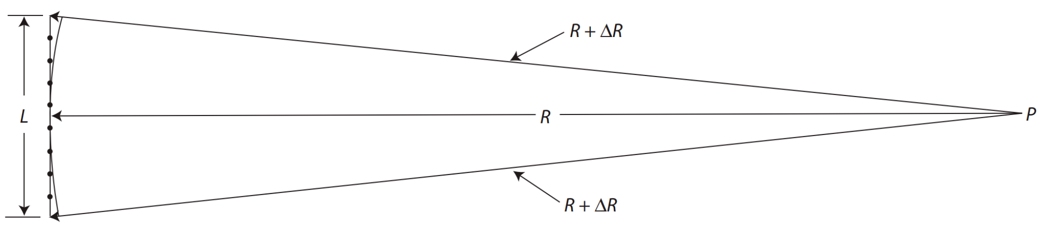


Figure 5: Distances Between Antenna Element and a Target [8]

Equation 9 from Stimson gives the phase delay in radians of an element at range R. The variable δ_n is the distance from the centre element of the synthetic array to an element. The further away an element is from the centre element, the greater the delay is. In Figure 6, ξ is visualized where the centre antenna element is in the middle of the x-axis and it is seen that the elements in the furthest from the centre element have the highest phase delay. The same figure also shows the phase delay decreases the further away a target is from the synthetic array. To correct for this delay, before the measurements are summed at the same distance, each element is multiplied by $e^{j\xi}$. Doing so will advance the phase without affecting the magnitude and focus the image.

A comparison between unfocused and focused SAR images can be seen below in Figure 7 with a synthetic array size of $\alpha = 67$. The images in Figure 7 were formed using the same recorded data of a scene. In the scene, there are two rectangular reflectors spaced 6m apart in cross-range and a wall that is approximately 11m from the radar.

$$\xi = \frac{2\pi}{\lambda} \frac{\delta_n^2}{R} \tag{9}$$

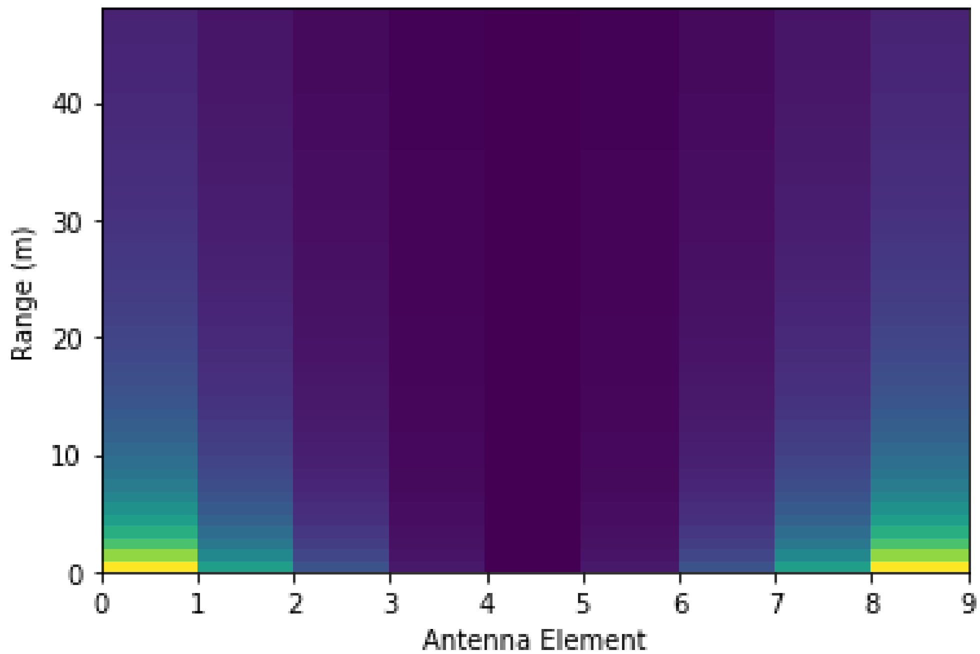


Figure 6: Plot of Synthetic Array Phase Delay

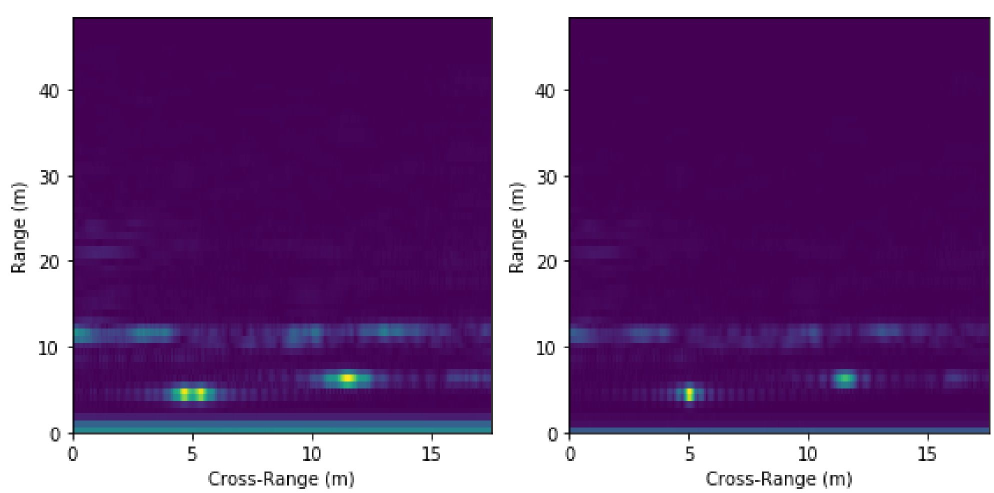


Figure 7: Comparison between unfocused and focused SAR images

2.4 Project Flow Chart

Figure 8 displays the logical flow of processes through which the project operates. Beginning at the commands taken from the user, the project operates through these steps until the platform has finished computing the image and the user can retrieve it off of the laptop.

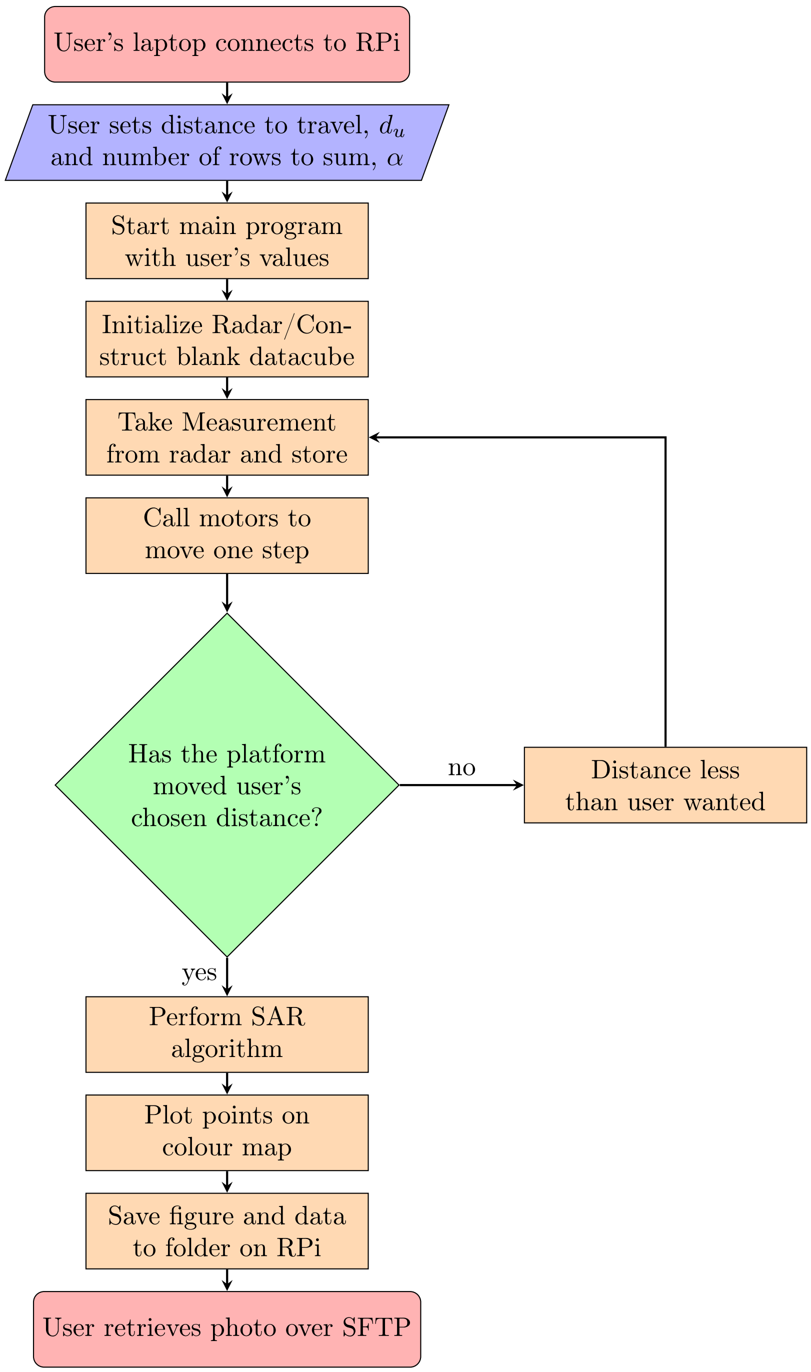


Figure 8: Project Flowchart

2.5 Equipment

The aim of the project is to build a mobile computing platform where an on-board computer communicates with the radar and motors producing a platform which can operate off of remote commands.

Figure 9: SAR Imaging Platform

2.5.1 Physical

- 1' x 1.5' plywood board
- 4x custom wheel brackets
- 4x 70mm aluminum wheels
- 2x stepper motors
- 2x stepper motor driver modules
- 12x female to female dupond cables
- custom aluminum wheel shaft
- 1/4" threaded bolt
- selfie stick
- radar mounting bracket
- Razor Blade Stealth 13" laptop
- Analog Devices EVAL-DemoRad
- 12V external battery pack
- microUSB to USB cable
- Raspberry Pi 4
- Raspberry Pi battery pack
- 32GB MicroSD card
- USB-C to USB-C cable

2.5.2 Software

- Raspbian OS - The operating system the mobile computer will utilize.

- Thonny IDE - The integrated development environment which will contain the code created.
- WinSCP - Third party software to enable SFTP connection between user's laptop and computer.

3 Architecture

This section is dedicated to explaining the hierarchy of the project as well as giving insight on how each module functions. Along with the descriptions of each module, how each module interacts with one another is also explained.

3.1 Description

The project consisted of a modular design where the major facets of the project were broken down into modules and the main program (*sarImage.py*) which controls and calls each segment when needed. The segments were broken down into four modules of initialization, motor control, measurement taking/image formation, and imaging handling. Figure 10 provides a look at the hierarchy and explains what task each module performs.

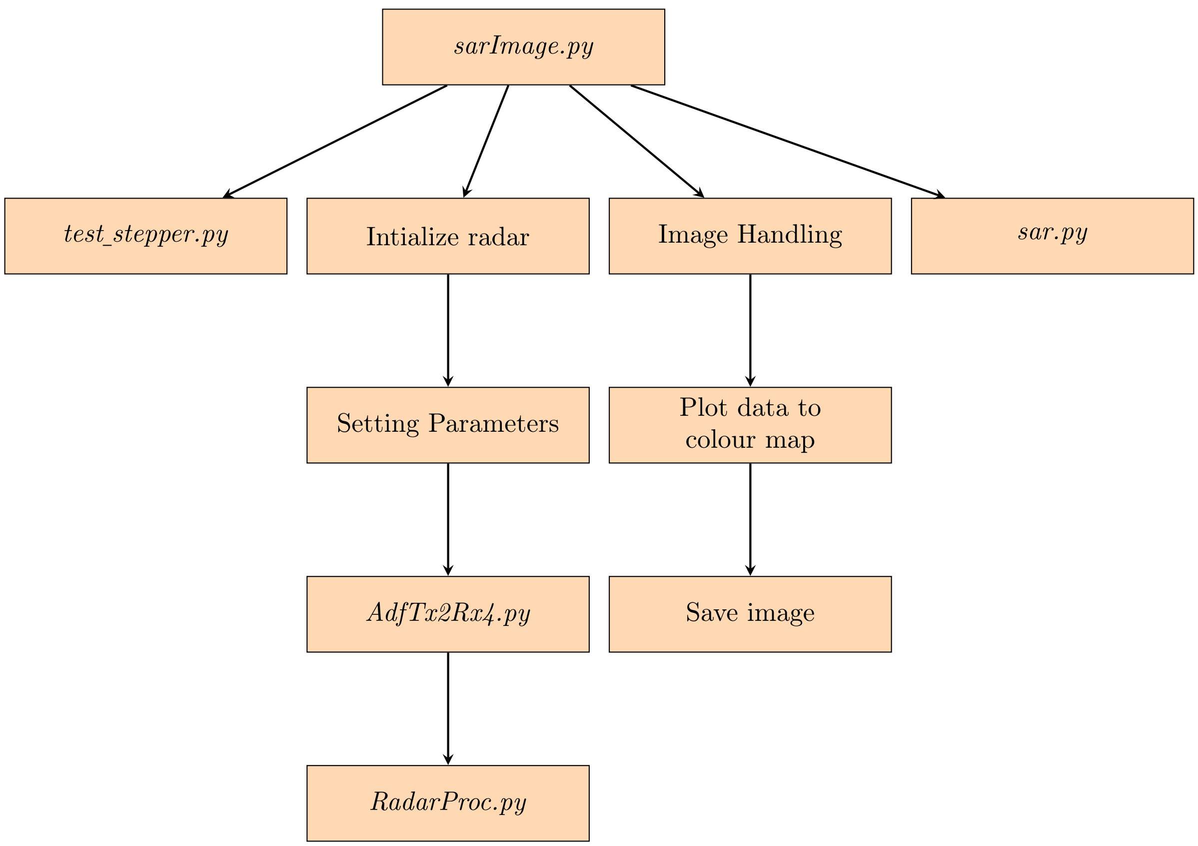


Figure 10: Code Hierarchy

3.2 SCP Connection

Prior to the execution of the main program, a connection is created between the Raspberry Pi and laptop as shown in Figure 11. This figure displays the interaction the user on the laptop and the Raspberry Pi have over the established connection. Using a third-party program, WinSCP, a connection is established between the two, allowing the user to access the files and command line of the Raspberry Pi. The established connections allowed the user the ability to send specific parameters such as distance to travel, and number of points to integrate allowing for slight customization depending on the environment being imaged. These parameters were entered on the command line at the same time that the user is telling the program to execute. Once the program had finished the same connection allowed the user to see the saved image stored on the Raspberry Pi.

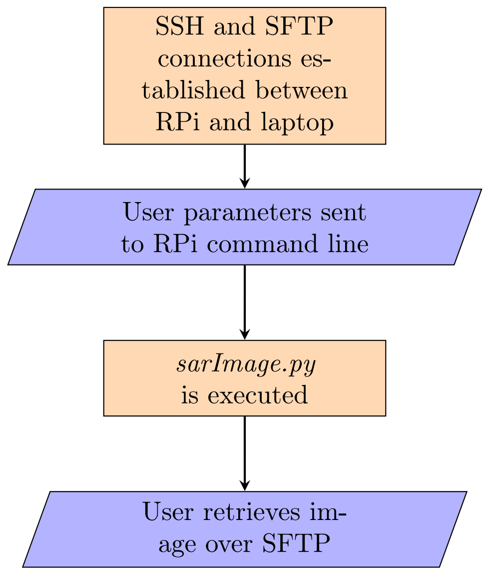


Figure 11: SCP Connection Flowchart

3.3 Radar Initialization

3.3.1 Parameters

The initialization section as shown in Figure 10 contains the setting up of the parameters of the radar. The DemoRad product did not inherently provide the data in the required form to perform SAR integration. The radar's parameters within the code were modified such that the radar gave the required data.

3.3.2 *AdfTx2Rx4.py*

This class was developed by Inras GmbH and creates parameters for the transmitter and receiver and initializes them to the values specified within this class [2].

3.3.3 *RadarProc.py*

This class was provided by Analog Devices and contains several useful functions for radar signal processing. The only function used from this class is to get the range profile of the DemoRad based on a Python dictionary called dRpCfg.

3.4 *test_stepper.py*

Figure 10 displayed the use of the stepper motor control as it was to give the platform the movement it required in small discrete steps. The code was established by initializing several of the GPIO pins on the Raspberry Pi as shown in Fig.12. Using these pins and several conditional statements the program was created such that calling the program would result in a single step from the motors. By calling the program rapidly with little delay between calls, the result would produce a rotational output which would move the platform. Combining the knowledge of the required movement step size as stated in Section 2.2 as well as the information on the motors' ability, the platform was able to move in discrete steps smaller than the maximum allowable value. Using the smallest motor step size of 5.625°(1/64th) [6], the given wheel radius size of 35mm given in the equipment list, and using equation 10 the motor step size of 3.43mm was found to be less than the maximum allowable step size of 4.66mm.

$$A_L = r\theta \tag{10}$$

This program was used by the main program by calling the stepper motor function each time a single step was required. This allowed for a logical flow from collecting one measurement from the radar and also taking one step of the platform.

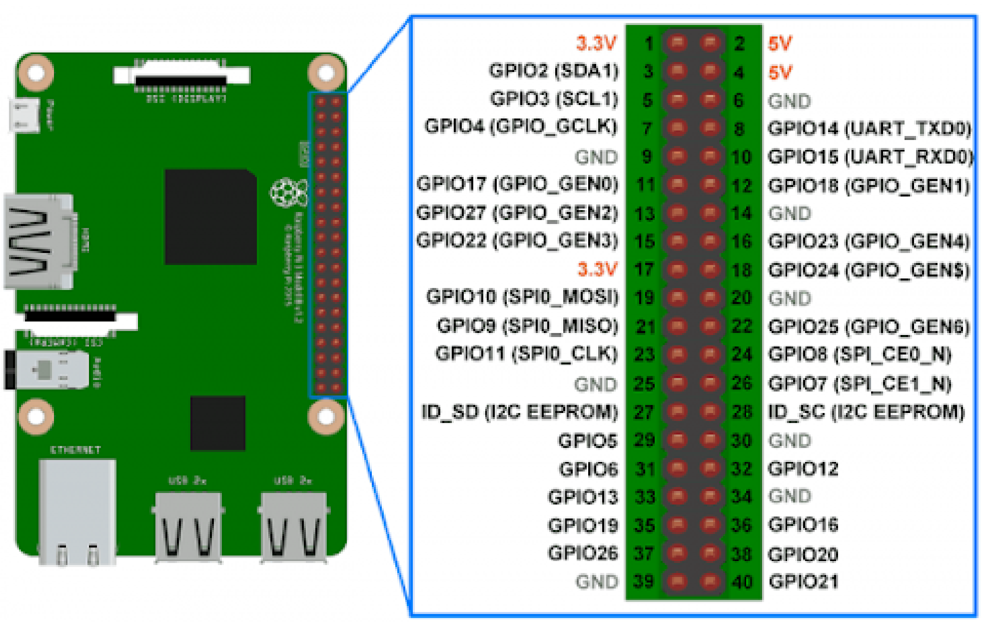


Figure 12: Raspberry Pi GPIO Pins

3.5 *sar.py*

This module was responsible for computing the SAR image using the algorithm described in Section 2 based on the data measured on the radar and the equations in Section 2. This program took in the data, and how many rows the user would like to sum α. To calculate the phase shift the range is needed so the variable k_f is needed to calculate the ξ. Based on the given data and user parameter this program implemented the strip-map algorithm and returned the user a new data cube with the integration having been completed. This data can then be plotted to produce an image.

3.6 Image Handling

Referencing Fig. 10 this section of the code took the returned data from *sar.py* and plotted it to a colour map and formatted the image with correct axes and titles. The completed image was then saved to a file with the time and date it was completed. After it is saved onto the Raspberry Pi the user can retrieve it off the laptop using the SCP connection stated in Section 3.2.

4 Testing

This section explains the testing done on the SAR imaging platform which included testing the requirements specified in the Statement of Requirements (SOR)[4]. All tests were conducted in a similar environment as seen in Figure 13 where metal reflective targets were set up in various configurations in front of the radar, while the platform would travel a set distance parallel to the targets. Figure 14 represents the SAR images produced where images exist as a heat map where the X-axis represents the horizontal movement the platform traversed, the Y-axis represents the down-range view from the radar, and imaged targets are represented by bright colours.

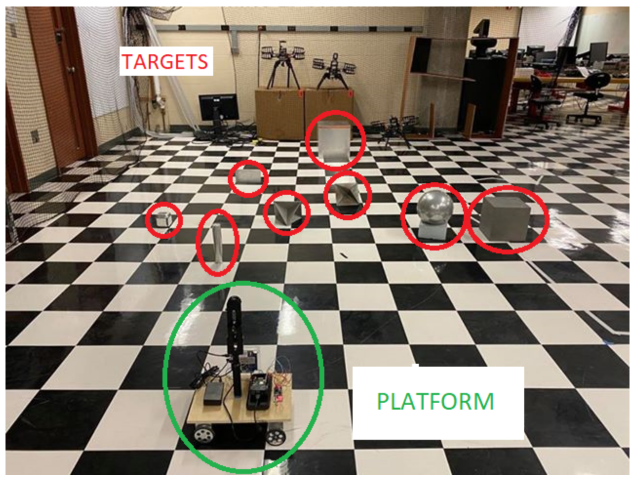


Figure 13: Sample Test Environment

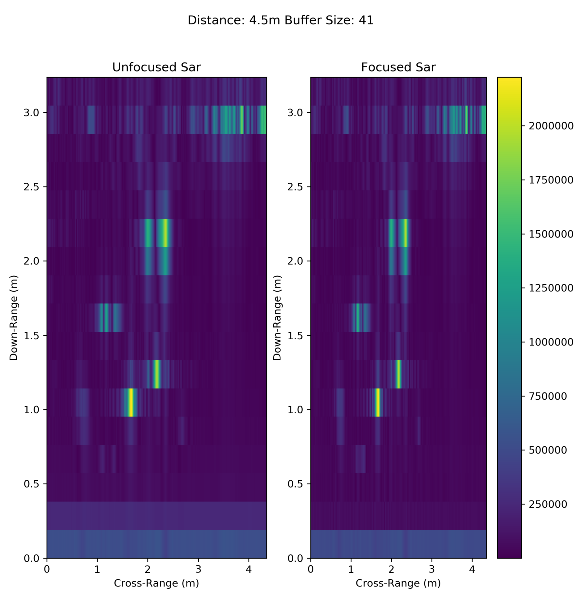


Figure 14: Sample SAR Image

4.1 Cross-range Resolution

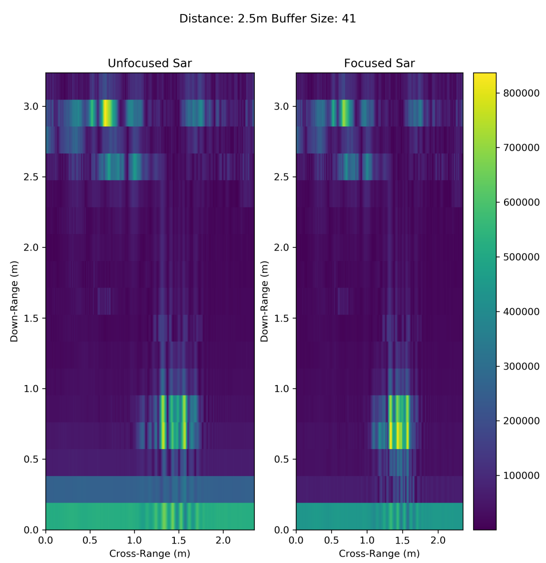
4.1.1 Description

In order to quantify the cross-range resolution a SAR image was produced of two targets closely spaced in cross-range, as depicted in Figure 15. When the targets were together they would appear as one target, and the cross-range resolution was defined as the distance at which the two targets were independently distinguishable. In this test the platform would travel a set 2.5m track and the targets (placed 1.20m away from the platform) were incrementally distanced apart starting together and ending at 35cm separation. The radar in this test was placed flat on the platform such that the look angle between the targets and the radar was 0°. While this was not a stated requirement for the project this test was crucial as the cross-range resolution was relevant for a surveillance platform.

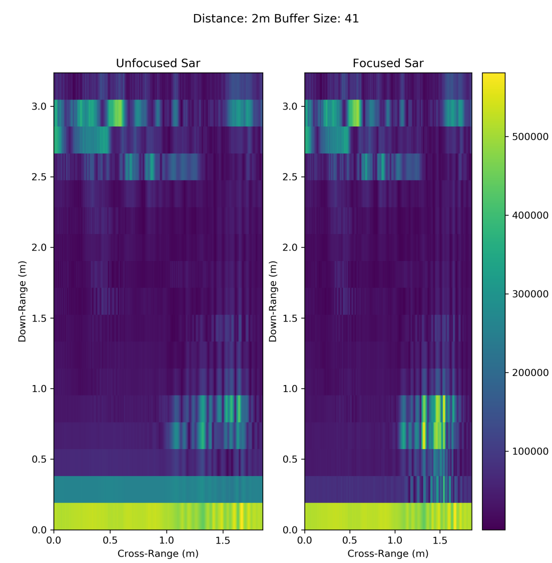
Figure 15: Cross-range Resolution Test Environment

4.1.2 Results

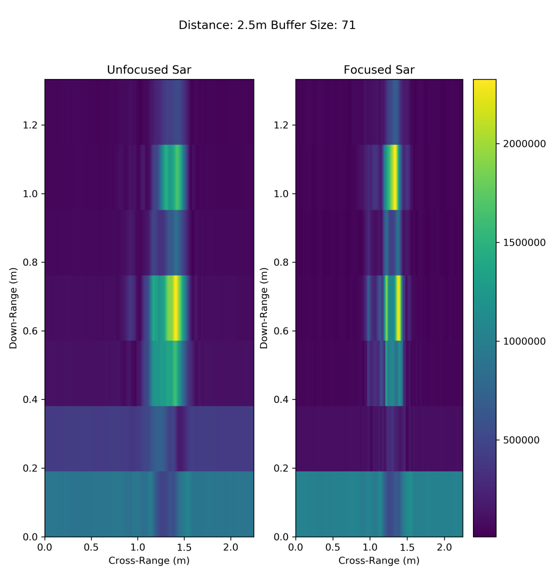
Figure 16 displays the four SAR images for various cross-range configurations ranging from 0cm separation up to 35cm. To verify the cross-range resolution the two targets were increasingly separated until there existed two noticeable independent bright spots on the image.



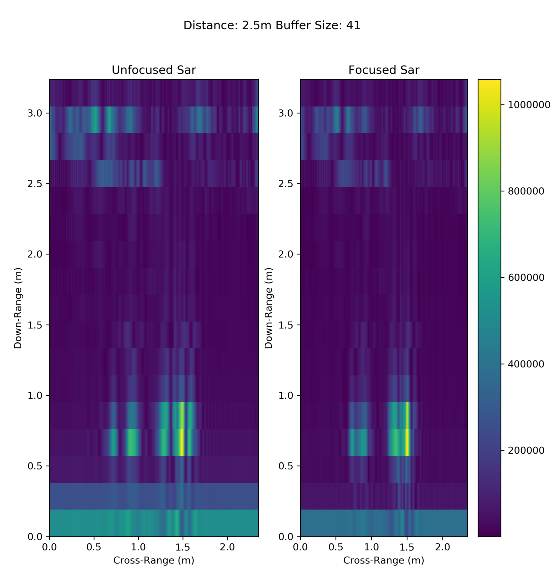
(a) CR SAR image with separation = 0 cm



(b) CR SAR image with separation = 1 cm



(c) CR SAR image with separation = 2 cm



(d) CR SAR image with separation = 35 cm

Figure 16: Cross-range SAR images

The DemoRad has a calculated cross-range resolution of 4.66mm as stated by equation 4 in section 2.2. With the results of the cross-range resolution test in Figure 16 it is shown that the resolution is not easily discernible. With the viewing of two plate targets, the results show that at small distances the resolution was not easily distinguishable. It is not until the separation was 35cm when two large returns separated by a null could be seen. At a distance of 2cm the separation is difficult to distinguish amongst the returns themselves. This test demonstrated that while the resolution is relatively small and theoretically distinguishable the practical results showed that with large targets the separation is hard to determine at small distances and virtually impossible for the user to discern two targets without prior knowledge of the scene.

4.2 Down-range Resolution

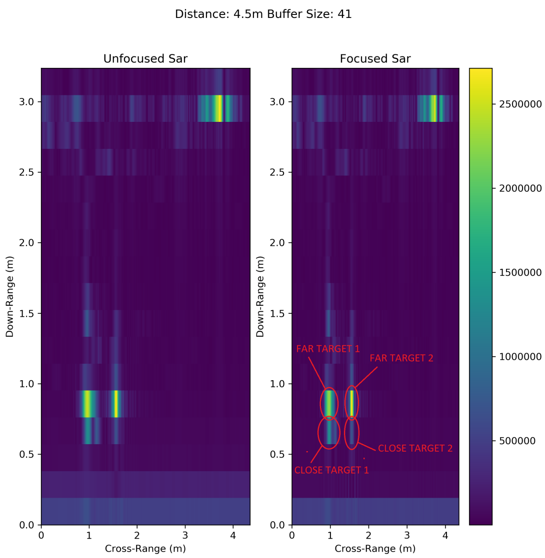
4.2.1 Description

The down-range resolution was measured similarly to the cross-range resolution. Using the same groups of two targets and a 2.5m track. In this test the DemoRad was placed on the platform providing a 0°look angle between the targets and the radar. The items were now spaced at various distances in down-range and the platform formed an image on them. In down-range, the distances between the targets varied from 30cm to 100cm. The purpose of this test was to see at which point two targets would become distinguishable from one another. This test was also not a requirement but was important for the surveillance platform's ability to see two targets down-range from one another.

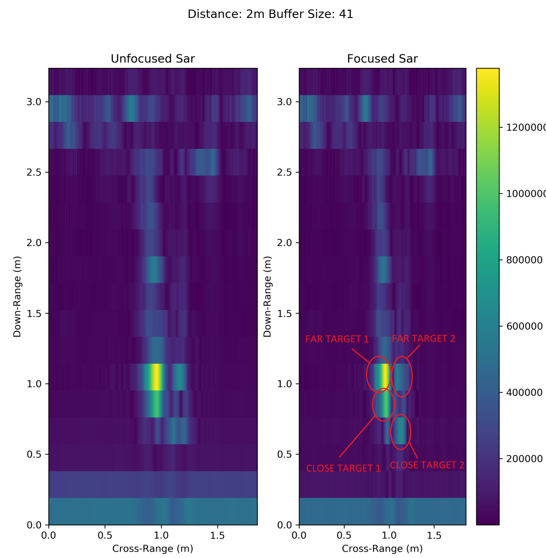
Figure 17: Down-range Resolution Test Environment

4.2.2 Results

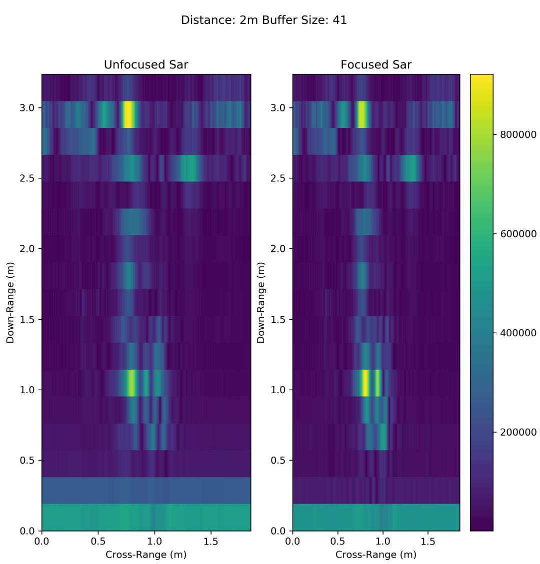
Figure 18 displays the four SAR images for various down-range configurations ranging from 30cm separation up to 100cm. To verify the down-range resolution the two targets were increasingly separated until there existed two noticeable independent bright spots on the image.



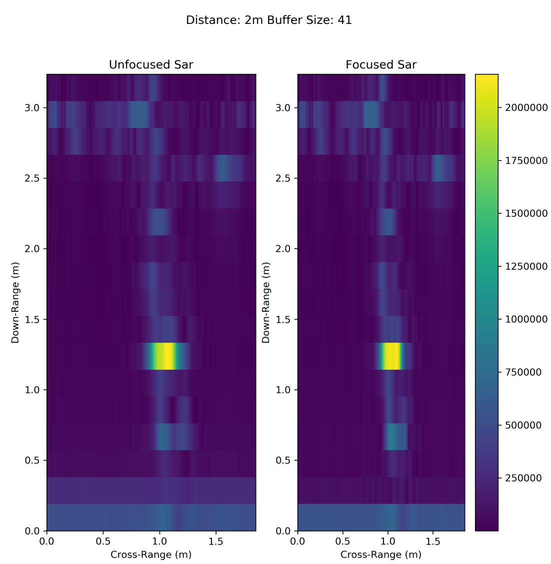
(a) DR SAR image with separation = 30 cm



(b) DR SAR image with separation = 50 cm



(c) DR SAR image with separation = 75 cm



(d) DR SAR image with separation = 100 cm

Figure 18: Down-range SAR images

The theoretical down-range resolution of the DemoRad is 50cm[2]. However, previous testing on the DemoRad found that one could not distinguish targets until they were separated by 75cm [2]. This test was able to confirm that as seen in the above figures. For the ranges of 30-75cm, there exists one long detection with a weaker front end and a stronger back end. This result remains consistent up until the separation was increased to 100cm. At a distance of 100cm, there exists a more obvious drop off in intensity between the forward and rear targets. This test helped quantify the DemoRad's capabilities as it confirmed the DemoRad's ability to distinguish a target in down-range.

4.3 Number of Integration Points

4.3.1 Description

The concept of the number of integration points, α, stated in Section 2.1 relates to the number of columns of data summed together in order to produce one column of pixels in the image. Reusing the data from a test where there were two strong targets in the scene (1.2m away from the radar) separated by a considerable distance and having the platform angled looking down on the scene and rotated sideways. Using this data the strongest return in the image was selected, while the number of columns being summed was increased each iteration. This value of the strongest return was plotted against the number of columns being summed. This test was designed to look for how many points of integration are required to produce a substantial return and what increasing the number of points does for the image. This test was not a requirement but aids in optimizing the platform as to not have the platform summing additional columns for no gain in the quality of returns and teasing the DemoRad's stated Field of View (FOV).

4.3.2 Results

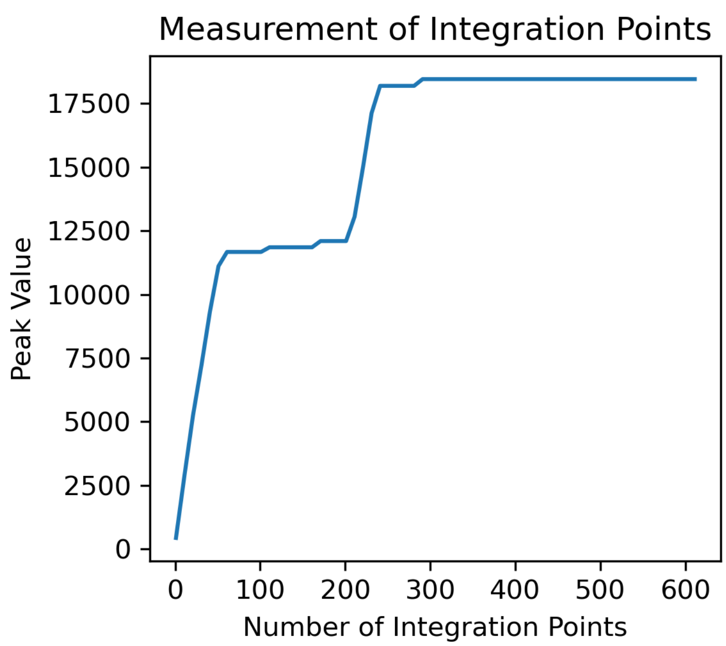


Figure 19: Number of Integration Points Test

The results in Figure 19 confirm that there is a limit on the number of integration points needed when forming an image. This limit comes from the fact that at a given distance to a target, the target is only within the radar's FOV for a limited distance. Once the target exits this FOV, summing additional points does not increase the strength of the return. This can be seen in the figure as at the beginning, increasing the number of columns increases the strength of the return considerably. As the number of summing points exceeds the distance that the target was within the FOV of the radar, the return begins to drop off characterized in the figure by the levelling off after the first increase. The figure however, shows a drastic increase again as the second target in the scene is included in the summation but levels off again once the size of the summation has included all of the second target.

Looking at the first increase due to the first target, the HPBW of the DemoRad can be tested. The first levelling off of the data comes after integrating 111 columns which at a step size of 3.43mm produces a distance of 387.3mm. Using trigonometry shown in equation 11 the HPBW for the DemoRad on its side is calculated to be 18.33°. Comparing this to the theoretical HPBW of the DemoRad in elevation of 15°[3] the HPBW is shown to be close in value. The slight difference is coming from the fact that the target does not exist as a single point but as a square target which will exist in the FOV for a longer period requiring more columns of integration before levelling off.

This test helped confirm the specifications of the radar's HPBW as well as understand that in order to optimize the platform, integration should only be done for the time that the target exists in the FOV. Doing so outside this range wastes computation or runs the risk combining the returns of two targets into one.

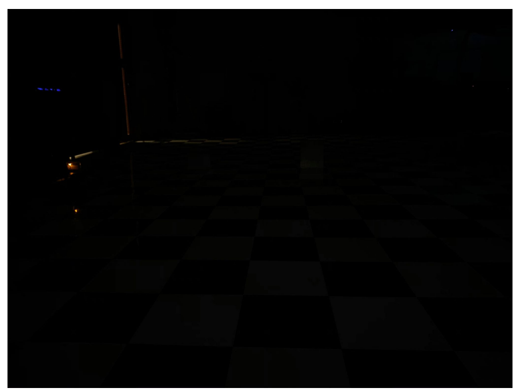
$$\theta_{HPBW} = 2tan^{-1}(\frac{\frac{dist.trav}{2}}{dist.totrgs}) \tag{11}$$

4.4 Unlit Conditions

4.4.1 Description

PR-03 specified that the platform would be able to image a target in lit and unlit conditions. This test was conducted by establishing a scene with several targets down-range from the platform with the lights off in the robotics laboratory. The scene was imaged in lit and unlit conditions with both images being retrieved by the user.

(a) Environment in Well-Lit Conditions

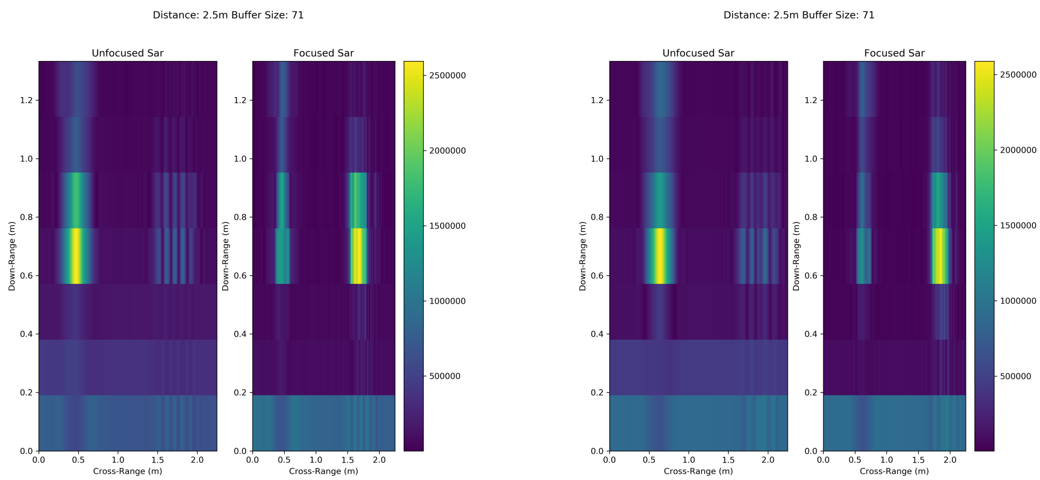


(b) Environment in Unlit Conditions

Figure 20: Lighting Conditions Test

4.4.2 Results

The above two figures demonstrate one of the strengths of SAR imaging. The above figures demonstrated that despite the lighting conditions, the images produced a nearly identical allowing SAR imaging to be used in different weather conditions or at night. This test confirmed the PR-03 requirement and confirmed one of the primary benefits of SAR imaging in a surveillance context.



(a) SAR Image in Well-Lit Conditions

(b) SAR Image in Unlit Conditions

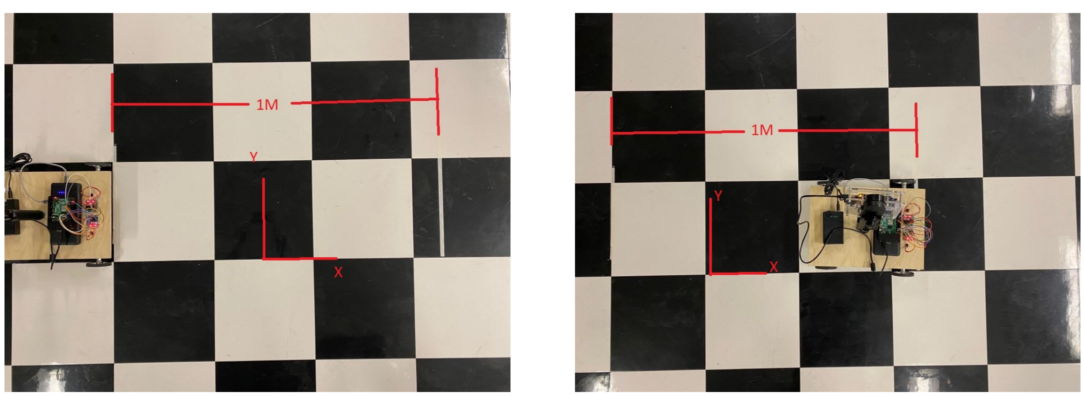
Figure 21: SAR Images from Lighting Conditions Test

4.5 Linear Movement

4.5.1 Description

FR-01 stipulated that the platform was to move in a single axis over a 1m distance. A test was conducted in the robotics laboratory where the platform was placed parallel to the tiles on the floor. The command was then sent to the platform for the platform to travel one metre to see how the platform moved.

4.5.2 Results



(a) Start Position of Platform's Movement Test

(b) End Position of Platform's Movement Test

Figure 22: 1m Linear Movement Test

The above figures demonstrate that along the 1m track that the platform moved linearly across the scene. This test confirmed that the platform was able to traverse the required distance travelling only in the positive X-direction and not travelling in the Y-direction.

4.6 Half Wavelength Movement

4.6.1 Description

PR-01 made the requirement that the platform was to move no more than a half wavelength (6.25mm) of displacement each time the stepper motors rotated. A test was established where a strip of 1m long tape was placed along the floor with the platform at one end. As the motors on the platform were set to make the smallest rotation possible, several steps were sent to the platform to see how many steps were required to travel 1m. This was test was finished when the platform consistently travelled the 1m at a set number of steps.

4.6.2 Results

The number of steps was determined to be 289 steps to travel one metre. This number ensured that the platform was moving 3.46mm per step, which is less than the required half-wavelength movement of 6.25mm and less than the worst-case scenario of 4.66mm. This test ensured that the requirement set out by PR-01 was satisfied.

4.7 Half Wavelength Measurement

4.7.1 Description

FR-03 was a requirement mandating that the radar take a measurement every half-wavelength of horizontal displacement. Based on the hierarchy of the code seen in Figure 10, it is clear that the platform takes the same number of measurements as the motor takes steps.

4.7.2 Results

Based upon the results of the test for the half-wavelength movement it was also seen that the radar was taking 289 measurements per metre of displacement. With this knowledge and the knowledge that the platform was moving 3.46mm per measurement, it was clear that FR-03 was satisfied.

4.8 Weight

4.8.1 Description

ImpR-03 was such that the platform with all devices was to weigh no more than 5lbs(2267g). To test the weight all the items were loaded onto the platform and were placed on a scale. This requirement was not critical for the project success but was a requirement such that a portable non-cumbersome platform was created.

4.8.2 Results

Based upon the above figure it can be seen that the platform weighs less than the max requirement of 2267g and weighs a total of 2218.9g as seen in the figure. This test confirmed the requirement and ensured that a portable platform was created.

Figure 23: Scale Reading

4.9 Radar Frequency

4.9.1 Description

PR-04 dictated that the DemoRad must operate in the IEEE K Band from 18-27 GHz. The DemoRad's documentation specifies that the minimum frequency is 23.9 GHz and the max frequency is 24.3 GHz [5]. The DemoRad was programmed to create SAR images with frequencies in this range which ensured that this requirement was met.

4.10 Wired Connections

4.10.1 Description

The requirement was to ensure that all connections on the platform were a wired connection. There is no communication between components on the platform. Referencing Figure 9 it can be seen that all connections between the RPi, the DemoRad, and the motors are a wired connection and thus this requirement was fulfilled.

4.11 Grayscale Image

This test was designed to fulfill FR-04 requiring the image to be in grey scale. This requirement was not a definite requirement due to the processing platform's ability to put the image into a plethora of colours. The use of grey scale was to ensure that the images were produced effectively.

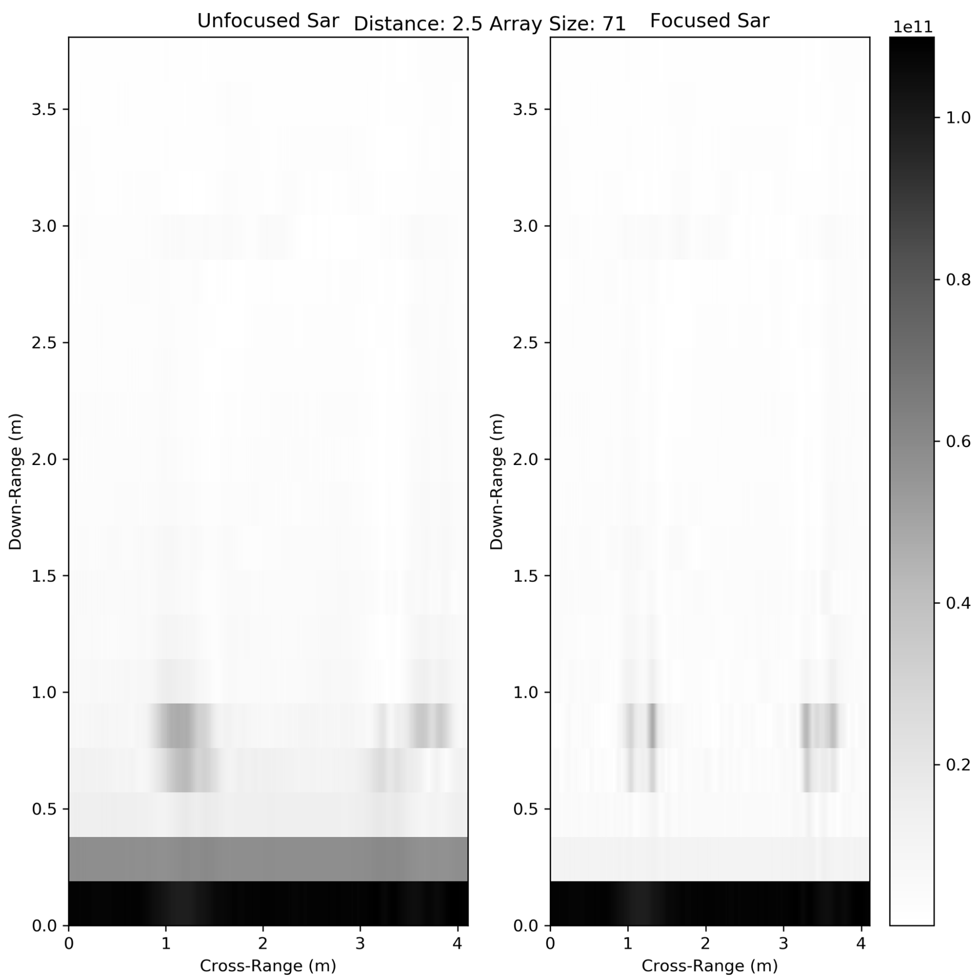


Figure 24: Grayscale SAR Image

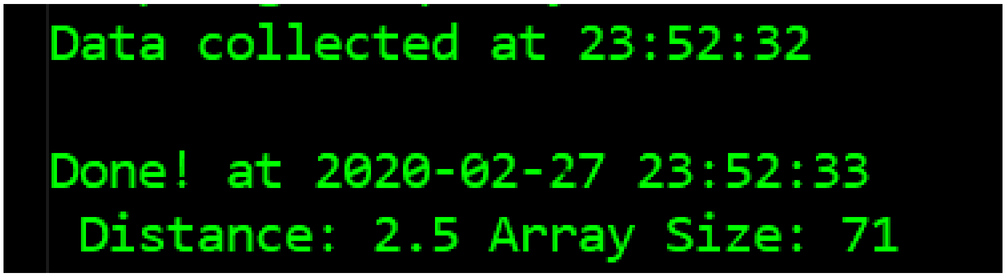


Figure 25: Code to Ensure 30s Formation

4.12 Formation in 30s

4.12.1 Description

PR-05 was a requirement forcing the image to be created in 30s or less. This requirement was to ensure that there was a reasonable delay between the accumulation of the data and the ability for the image to be retrieved by the user. To ensure this requirement was met the code was altered slightly so that the program would send the user statements to know when all the data had been collected and when the image was formed. The data was based on a scene with two targets, and performing a summation of 71 measurements per column of pixels over a 1m displacement. Based on the figure below it can be seen that from the time the platform has collected the data to having the image formed is one second ensuring that this requirement was met and the fact that the time to produce the image is much less than requirement ensures that this requirement will be met for any scenario within of the scope of the project.

4.13 Battery Requirement

4.13.1 Description

As per the SOR, the platform was designed such that all electronic components operated off of a battery so the platform was able to be mobile and for the platform to operate for at least 30 minutes[4]. This was achieved by having a battery to satisfy the RPi requirements and a separate one for the DemoRad.

The RPi has a power requirement of 5V 3A [7] which is satisfied by the 20800 mAh battery that the RPi is sitting on in Figure 9. With the consumption of 3 A by the RPi, this ensured that the RPi with peripherals would last for just less than 7 hours drawing full power, much larger than the 30-minute requirement.

The DemoRad's power requirements were 12 V pulling a maximum current of 290 mA [5]. The battery seen on the far left in Figure 9 handled this as it was able to supply 12V at 3000 mAh [1]. With the DemoRad at full consumption, this meant the battery could last for around 10 hours and satisfying the 30-minute requirement.

With these two batteries, the platform was able to be mobile and these batteries satisfied the requirements set out by FR-05 and PR-02.

4.14 Transmission of Image

4.14.1 Description

The IR-02 requirement was such that a PC would be able to retrieve the image after the RPi had completed forming it. Using a Secure File Transfer Protocol (SFTP) connection between the RPi and the laptop the user was able to navigate and retrieve the completed photos in the RPi's figure folder.

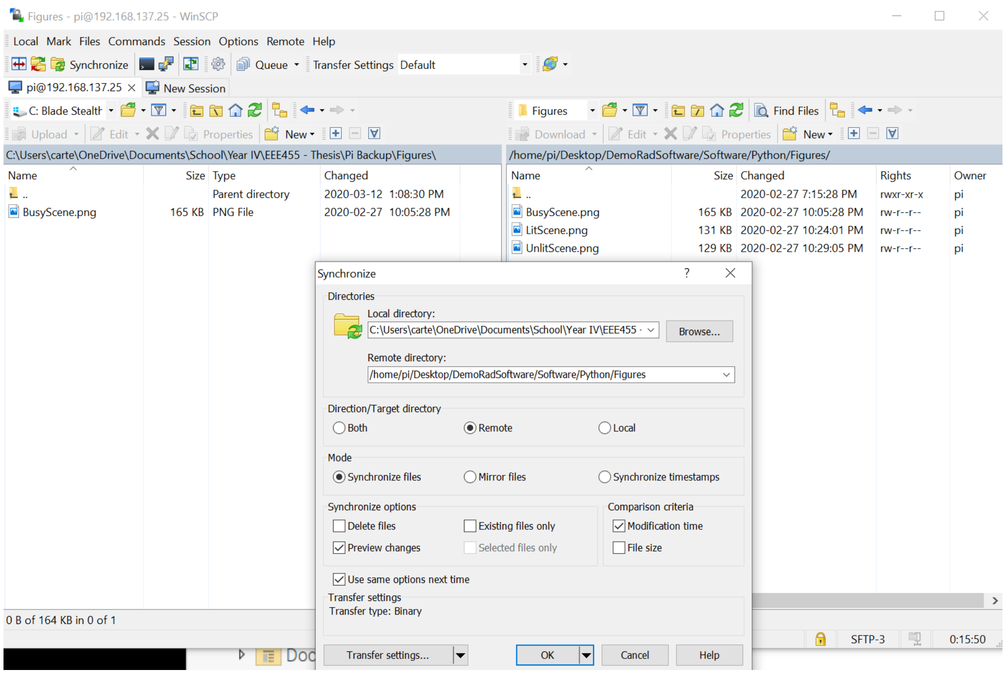


Figure 26: User retrieving SAR Image

4.15 Remote Control

4.15.1 Description

IR-03 required that the platform be started by a command from the external laptop. A test was established where the platform was disconnected from any monitor and without any input devices (mouse, keyboard) plugged into it. Using the connection between the RPi and the laptop, the RPi's command line was told to execute the main program for a one-metre displacement. The platform operated and began to move confirming that the connection had sent the command.

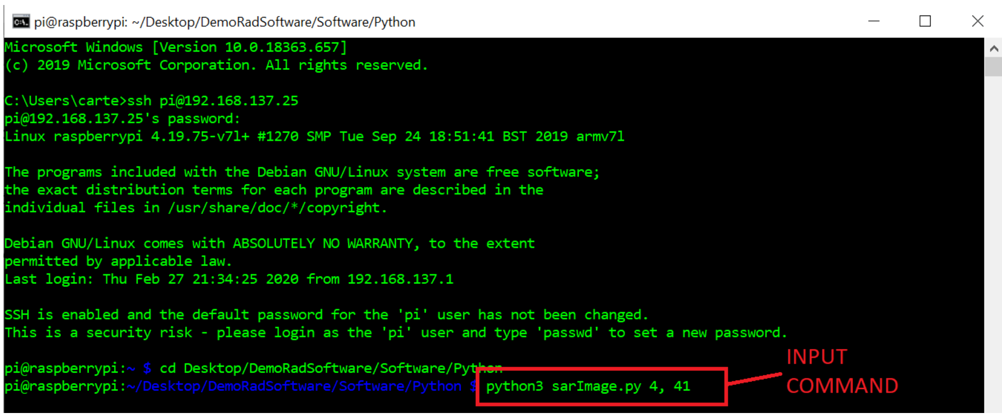


Figure 27: Calling *sarImage.py* from Command Line Over SSH

4.16 On-board Processing

4.16.1 Description

ImpR-01 and ImpR-02 were requirements that the imaging processing and motor control were to be done on the RPi and not sent to the external laptop. Based on Figure 10 it can be seen that the program held on the RPi handled both the image formation and the control of the motors. This handling of those two features on the RPi ensured that the RPi was the one in control of those segments and not the external laptop.

4.17 Presentation of Results

4.17.1 Functional Requirements

Requirement	Description	Results
FR-01	The platform will move in a single axis on a track that shall be 1m long.	Satisfied
FR-02	The system will display the produced image on a 4in. screen of resolution 800x480 pixels mounted to the platform.	Unsatisfied
FR-03	The radar shall take a measurement at least every $\frac{1}{2}$ wavelength.	Satisfied
FR-04	The project shall produce a greyscale synthetic aperture radar image.	Satisfied
FR-05	Batteries will power all devices ensuring that the platform is mobile.	Satisfied

4.17.2 Performance Requirements

Requirement	Description	Results
PR-01	The DC motor shall produce incremental movement providing no more than $\frac{1}{2}$ wavelength translation distance.	Satisfied
PR-02	The batteries shall provide at least half an hour of continuous use to the mobile computing platform and the radar.	Satisfied
PR-03	The platform shall produce an image that will appear the same regardless of visibility in the visible light spectrum.	Satisfied
PR-04	The radar shall operate in the IEEE K Band.	Satisfied
PR-05	The platform shall form an image in no longer than 30 seconds.	Satisfied

4.17.3 Interface/Simulation/Implementation Requirements

Requirement	Description	Results
IR-01	All connections shall be wired. Interfacing between components on the platform will not be wireless.	Satisfied
IR-02	The platform will have the capability to transmit an image via a Wi-Fi connection to a PC.	Satisfied
IR-03	Another PC shall be able to send a 'start' command to the mobile platform.	Satisfied
SimR-01	A simulated scene of targets (such as vehicles or people) shall be constructed to allow testing of the imaging platform in well lit and unlit situations.	Satisfied
ImpR-01	All processing of the imaging shall be done on the mobile computing platform.	Satisfied
ImpR-02	All control of the motors shall be done on the mobile computing platform.	Satisfied
ImpR-03	The platform shall weigh no more than 5 lbs. to ensure portability.	Satisfied

4.17.4 Comments

The requirement FR-02 was abandoned in the project due to budget and its inadequacy in this scenario. The budget reason came from the fact that after ordering a Raspberry Pi, motors, wheels, and batteries there was not enough left in the budget to purchase a quality image that could function on the RPi. Displays that were made for the Raspberry Pi also use all of the General Purpose Input and Output (GPIO) ports on the Raspberry Pi. This would have made us unable to use stepper motors which were much more important to the functioning of the platform than having a display on the platform. The inadequacy reasons for the screen came from the fact that the platform was already to be interfaced with a separate laptop to view the image. The existence of a screen on the platform while trying to build a remote platform, where you can already see the image from a distance did not fit practically within this scenario and this requirement was therefore disregarded.

5 Discussion

The project's aim of producing a SAR imaging platform using an FMCW radar was a success and met the project's intended target. The practicality of using the produced SAR imaging platform, however, was not a success as the platform itself is not a useful platform. The platform is made from relatively cheap materials and these materials hurt the usefulness as the platform is unable to be used in anything other than our narrow testing situations. The motors themselves prove to be an issue as the motors travelled at slow speeds and would occasionally not travel at a consistent speed. This was an issue as in a project where the movement of the platform is critical, inconsistencies would drastically alter the results. As well the motors slow speed came from the weight of the platform and this meant the platform would only work on flat floors. When the platform was tested in an environment with any sort of incline or grooves in the floor, the platform would not move and remain stationary. This meant the performance of the platform was poor in any situation that was not ideal. The usefulness of the images produced is another area of concern as the platform was only able to be tested on objects close to the platform due to the limited space in the ideal environment. The use of the platform at long distances on objects a user would be interested in imaging is unknown due to the poor quality of the items making up the platform. While an image is produced , the image's advantage over what an optical image could produce was never fully demonstrated. While being a successful proof of concept on FMCW SAR images, the platform's images fail to provide anything meaningful that the user would not already be able to see.

6 Conclusion

The purpose of this project was to develop a SAR imaging platform using an FMCW radar. This goal was met and the platform was produced meeting almost all of the specified requirements. The platform is interfaced with an external laptop and runs off of a given command allowing the user to be away from the platform as theorized in the hypothetical scenario. The platform contained all the necessary components to produce a SAR image on board for the user to view remotely. The images produced, while not completely accurate, were largely limited by the DemoRad platform's specifications and not the types of technology utilized. With this platform, the concept of FMCW SAR imaging was demonstrated to be viable and the project goals were accomplished.

References

[1] Amazon. *Talent Cell Rechargable Battery Specifications*. URL: https://www.amazon.ca/TalentCell-Rechargeable-3000mAh-Lithium-External/dp/B01M7Z9Z1N/ref=sr_1_5?keywords=3000mah+12V&qid=1583684467&sr=8-5 (visited on 03/08/2020).

[2] OCdt Taylor Curran and OCdt Jin Park. "DID-07 – Detailed Design Document: Man-portable Radar System". In: (2019), p. 46.

[3] *EVAL-DEMORAD Evaluation Board | Analog Devices*. URL: https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-demorad.html#eb-documentation (visited on 03/04/2020).

[4] OCdt Carter Green and NCdt Royce Burningham. *DID03-Statement of Requirements*. 2020.

[5] Andreas Haderer. *DemoRad (Hardware User Manual)*. 2017.

[6] Longruner. *Longruner 5pcs Geared Stepper Motor 28byj 48 Uln2003 5v Step*. URL: http://www.longrunerpro.com/a/Products/20191129/248.html (visited on 04/04/2020).

[7] RaspberryPi. *FAQs*. URL: https://www.raspberrypi.org/documentation/faqs/#pi-power (visited on 03/08/2020).

[8] George W. Stimson et al. *Stimson's Introduction to Airborne Radar*. 3rd ed. Electromagnetics and Radar. Version Number: 3. SciTech Publishing, 2014. ISBN: 978-1-61353-022-1.

Appendix A - Main Program Code

```
# Authors Royce Burningham & Carter Green

import Class.Adf24Tx2Rx4 as Adf24Tx2Rx4
import Class.RadarProc as RadarProc
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
import test_stepper as step
import RPi.GPIO as GPIO
import sar
import datetime as datetime
import sys

STEP_DISTANCE = 3.4636e-3
c0 = 3e8

# Setup Connection
Brd = Adf24Tx2Rx4.Adf24Tx2Rx4()
Brd.BrdRst()

# Configure Receiver
Brd.RfRxEna()
TxPwr = 100

# Configure Transmitter (Antenna 0 - 3, Pwr 0 - 31)
Brd.RfTxEna(1, TxPwr)

# Configure Up-Chirp
dCfg = {
    "fs": 1.0e6,
    "fStrt": 23.9e9,
    "fStop": 24.3e9,
    "TRampUp": 260 / 1.0e6,
    "Tp": 300 / 1.0e6,
    "N": 256,
    "StrtIdx": 0,
    "StopIdx": 1,
    "MimoEna": 0,
    "Frms": 1,
}

Brd.RfMeas("Adi", dCfg)

kf = (dCfg["fStop"] - dCfg["fStrt"]) / dCfg["TRampUp"]
fs = Brd.RfGet("fs")
FuSca = Brd.Get("FuSca")

dRpCfg = {
    "RemoveMean": 1,
    "FFT": 2 ** 9,
    "FuSca": FuSca,
    "fs": fs,
```

```python
    "kf": kf,
    "RMin": 0,
    "RMax": 4,
    "dB": 0,
    "Abs": 0,
    "Ext": 1,
    "NrFrms": 128,
}

Proc = RadarProc.RadarProc()

Range = Proc.GetRangeProfile("Range")
Proc.CfgRangeProfile(dRpCfg)

fStrt = Brd.Adf_Pll.fStrt
fStop = Brd.Adf_Pll.fStop
TRampUp = Brd.Adf_Pll.TRampUp
n = np.arange(int(dCfg["N"]))
Range = Proc.GetRangeProfile("Range")

# Configure SAR Features
if len(sys.argv) < 2:
    points = 166
    sum_size = 9
else:
    distance = float(sys.argv[1])
    points = int(distance / STEP_DISTANCE)
    if int(sys.argv[2]) % 2 == 0:
        sum_size = int(sys.argv[2]) + 1
    else:
        sum_size = int(sys.argv[2])

DataCube = np.zeros((dCfg["N"], points), dtype="int16")

# Collect Data
for i in range(0, points):
    Data = Brd.BrdGetData()
    DataCube[:, i] = np.flip(Data[:, 0])
    step.step(8)
    # print(i)

# Process Data
Out = sar.sar(DataCube, sum_size, window=False) / FuSca
Out2 = sar.focused_sar(DataCube, sum_size, kf, window=False) / FuSca

X, Y = np.meshgrid(np.arange(points - sum_size), Range)
lam = c0 / (dCfg["fStrt"] + dCfg["fStop"]) / 2
CR = X * lam * Y / (2 * STEP_DISTANCE * sum_size)

# Plot SAR Image
fig, ax1 = plt.subplots(1, 2, figsize=(8.5, 8.5))

beep = ax1[0].pcolormesh(
    X * lam / (2 * STEP_DISTANCE * sum_size),
```

```
    Y,
    np.abs(Out)[0 : len(Range), 0 : points - sum_size],
    cmap=cm.Greys,
)
fig.colorbar(beep, ax=ax1[1])
ax1[0].set_xlabel("Cross-Range (m)")
ax1[0].set_ylabel("Down-Range (m)")
ax1[0].set_title("Unfocused Sar")

ax1[1].pcolormesh(
    X * lam / (2 * STEP_DISTANCE * sum_size),
    Y,
    np.abs(Out2)[0 : len(Range), 0 : points - sum_size],
    cmap=cm.Greys,
)
ax1[1].set_xlabel("Cross-Range (m)")
ax1[1].set_ylabel("Down-Range (m)")
ax1[1].set_title("Focused Sar")

fig.suptitle("Distance: {} Array Size: {}".format(sys.argv[1], sys.argv[2]))
fig.tight_layout()

# Save Plot
fig_path = "/home/pi/Desktop/DemoRadSoftware/Software/Python/Figures/"
today = datetime.datetime.now().strftime("%Y-%m-%d_%H:%M:%S")
plt.savefig(fig_path + today + ".png", dpi=300)

# Save Data
my_path = "/home/pi/Desktop/DemoRadSoftware/Software/Python/SavedData/"
np.save(my_path + today + ".npy", DataCube)

# Set GPIO ports to 0
GPIO.cleanup()

# Let user know image and data are saved
print(
    "\nDone! at "
    + today
    + "\n Distance: {} Array Size: {}".format(sys.argv[1], sys.argv[2])
)
```

Appendix B - SAR algorithm code

```
import numpy as np
import scipy.signal as sg

D0 = 3.4636e-3  # Gathered experimentally
lam = 3e8 / (24.1e9)


def sar(Data, k, window=True):
    if window:
        Win = np.hanning(Data.shape[0])
        ScaWin = Win.sum()
        Win = np.tile(Win, (Data.shape[1], 1)).T
        Data = Data * Win / ScaWin

    output = np.zeros((Data.shape[0], Data.shape[1] - k), dtype="float64")
    for i in range(0, Data.shape[1] - k):
        output[:, i] = Data[:, i : i + k].sum(axis=1)

    return np.fft.rfft(output, axis=0)


def sar2(Data, k, window=True):
    if window:
        Win = np.hanning(Data.shape[0])
        ScaWin = Win.sum()
        Win = np.tile(Win, (Data.shape[1], 1)).T
        Data = Data * Win / ScaWin

    h = np.ones((1, k))
    output = sg.convolve(h, Data, mode="valid")

    return np.fft.rfft(output, axis=0)


def focused_sar(Data, k, kf, window=True):
    b = k // 2
    d = np.abs(np.arange(-b, b + 1, dtype=int) * D0)
    Range = np.fft.rfftfreq(Data.shape[0], 1e-6) * 3e8 / (2 * kf)
    X, Y = np.meshgrid(d, Range)
    xi = np.divide(2 * np.pi * X ** 2, (lam * Y), out=np.zeros_like(X), where=Y != 0)
    phase_corr = np.exp(1j * xi)

    if window:
        Win = np.hanning(Data.shape[0])
        ScaWin = Win.sum()
        Win = np.tile(Win, (Data.shape[1], 1)).T
        Data = Data * Win / ScaWin

    data_freq = np.fft.rfft(Data, axis=0)

    output = np.zeros((data_freq.shape[0], data_freq.shape[1] - k), dtype="complex128")
    for i in range(0, data_freq.shape[1] - k):
```

```
        output[:, i] = (data_freq[:, i : i + k] * phase_corr).sum(axis=1)
    return output
```

Appendix C - Stepper Motor Code

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

halfstep_seq = [
    [1,0,0,0], #N
    [1,1,0,0], #NE
    [0,1,0,0], #E
    [0,1,1,0], #SE
    [0,0,1,0], #S
    [0,0,1,1], #SW
    [0,0,0,1], #W
    [1,0,0,1] #NW
]

control_pins = [7, 11, 13, 15]
control_pins2 = [31, 33, 35, 37]

for pin in control_pins:
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, 0)

for pin in control_pins2:
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, 0)

def step(n):
    for i in range(n):
        for halfstep in range(len(halfstep_seq)):
            for pin in range(4):
                GPIO.output(control_pins[pin], halfstep_seq[halfstep][pin])
                GPIO.output(control_pins2[pin], halfstep_seq[halfstep][pin])
            time.sleep(.001)

if __name__ == "__main__":
    for num in range(288):
        step(8)
#GPIO.cleanup()
```

Appendix D - DemoRad Coherence Testing

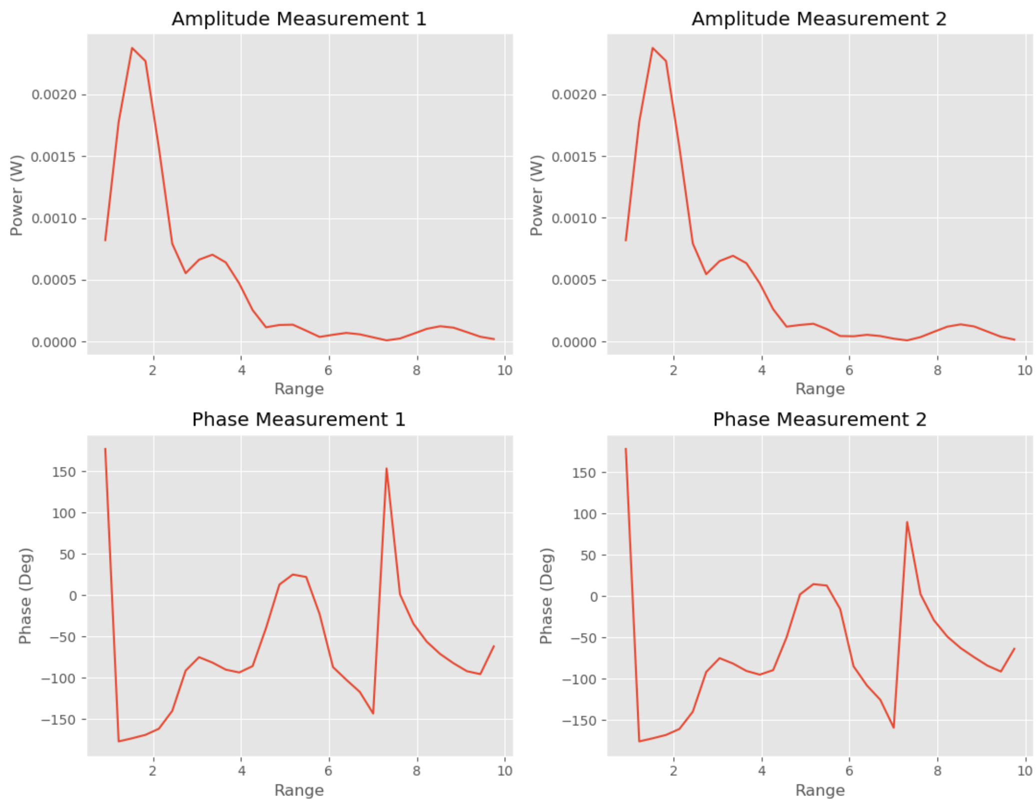


Figure 28: Testing Coherency of the Radar

SAR imaging requires that the individual measurements must be coherent with each other. If they were not then adding consecutive measurements to each other would cause interference and would not produce an image. To test this a reflective target was set in front of the DemoRad approximately 2m away. The DemoRad then took 2 measurements and compared the magnitude and phase of the two measurements which can be seen in figure 28. Repeating this test showed that the DemoRad is coherent between measurements. Noise adds random phase to the reading but when summing multiple returns the random phases will likely destructively interfere with each other and disappear.